数学建模与数学实验

最短路问题

图 论 的 基 本 概 念

一、 图 的 概 念

1. 图的定义

2. 顶点的次数

3. 子图

二、 图 的 矩 阵 表 示

1. 关联矩阵

2. 邻接矩阵

图的定义

定义　有序三元组$G=(V,E,\Psi)$称为一个**图**,如果:

[1] $V=\{v_1,v_2,\cdots,v_n\}$是有限非空集，V称为**顶点集**，其中的元素叫图 G 的**顶点.**

[2] E 称为**边集**，其中的元素叫图 G 的**边.**

[3] Ψ 是从边集 E 到顶点集 V 中的有序或无序的元素偶对构成集合的映射，称为**关联函数.**

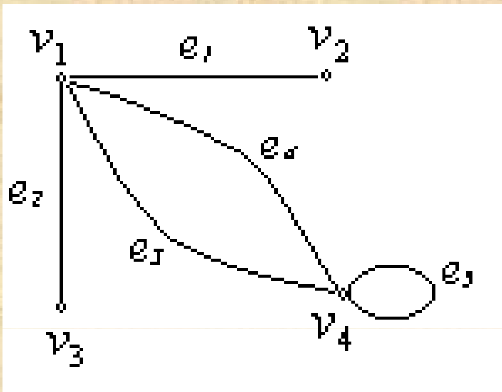
例1　设 $G=(V,E,\Psi)$，其中

$V=\{v_1,v_2,v_3,v_4\}$，

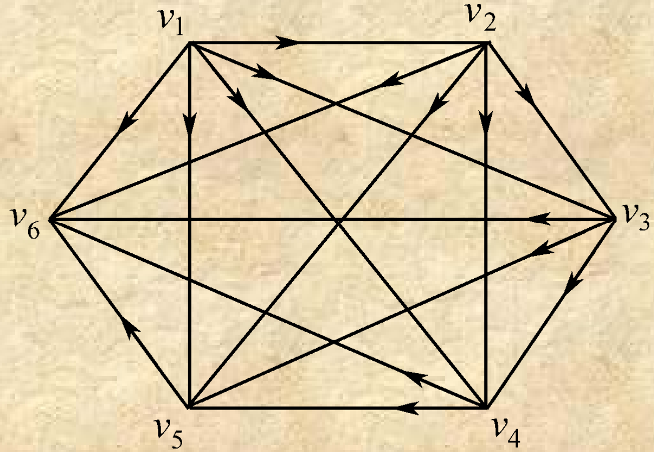
$E=\{e_1,e_2,e_3,e_4,e_5\}$，

$$\Psi(e_1)=v_1v_2,\Psi(e_2)=v_1v_3,\Psi(e_3)=v_1v_4,\Psi(e_4)=v_1v_4,\Psi(e_5)=v_4v_4.$$

G 的图解如图

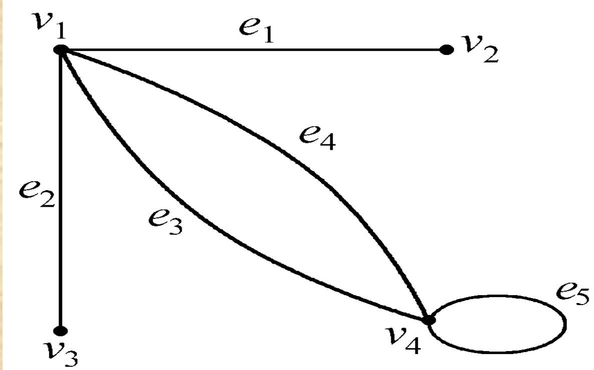


定义 在图 G 中，与 V 中的有序偶$(v_i,\ v_j)$对应的边 e，称为图的**有向边**（或弧），而与 V 中顶点的无序偶 v_iv_j 相对应的边 e，称为图的**无向边**.每一条边都是无向边的图，叫**无向图**；每一条边都是有向边的图，称为**有向图**；既有无向边又有有向边的图称为**混合图**.



定义 若将图 G 的每一条边 e 都对应一个实数 $w(e)$，则称 $w(e)$为边的**权**，并称图 G 为**赋权图**.

规定用记号 ν 和 ε 分别表示图的顶点数和边数.



常用术语：

（1）端点相同的边称为**环**.

（2）若一对顶点之间有两条以上的边联结，则这些边称为**重边**.

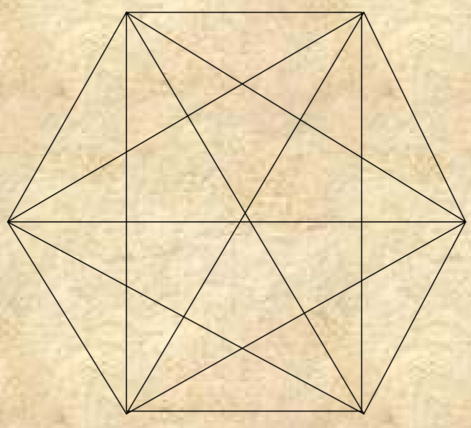
（3）有边联结的两个顶点称为**相邻的顶点**，有一个公共端点的边称为**相邻的边**.

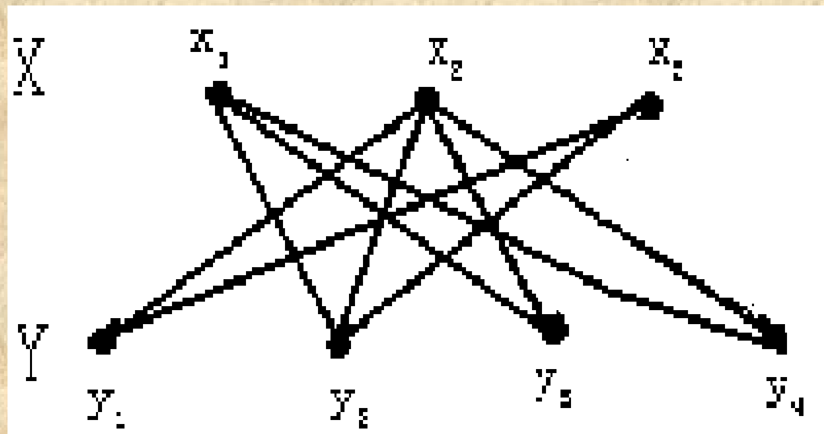
（4）边和它的端点称为互相**关联**的.

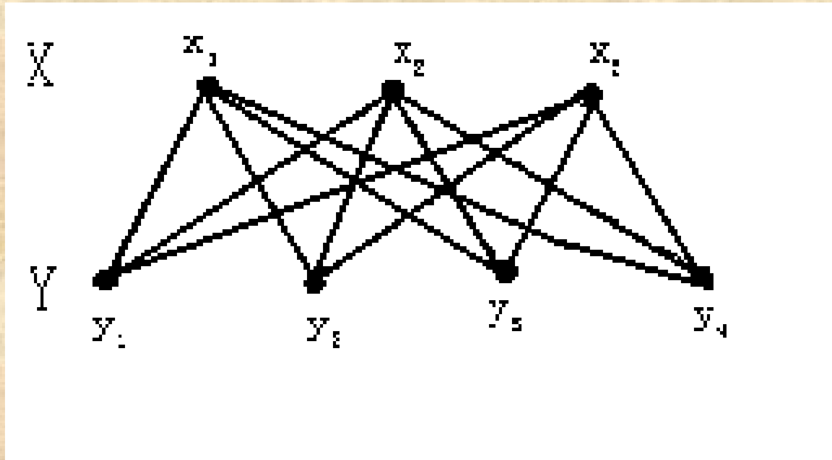
（5）既没有环也没有重边的图，称为**简单图**.

（6）任意两顶点都相邻的简单图，称为**完备图**，记为 K_n，其中 n 为顶点的数目.

(7)若 $V=X\cup Y$，$X\cap Y=\Phi$，且 X 中任两顶点不相邻，Y 中任两顶点不相邻，则称 G 为**二元图**；若 X 中每一顶点皆与 Y 中一切顶点相邻，则 G 称为**完备二元图**，记为 $K_{m,n}$，其中 m,n 分别为 X 与 Y 的顶点数目.



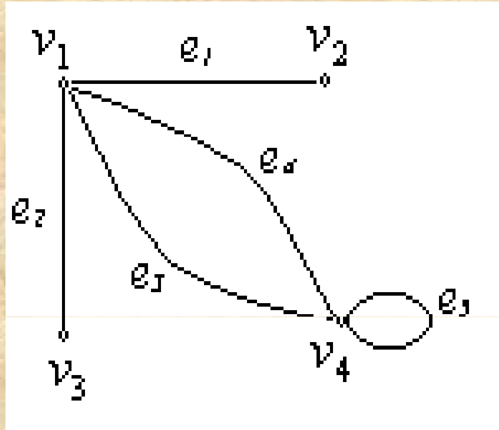

X
Y

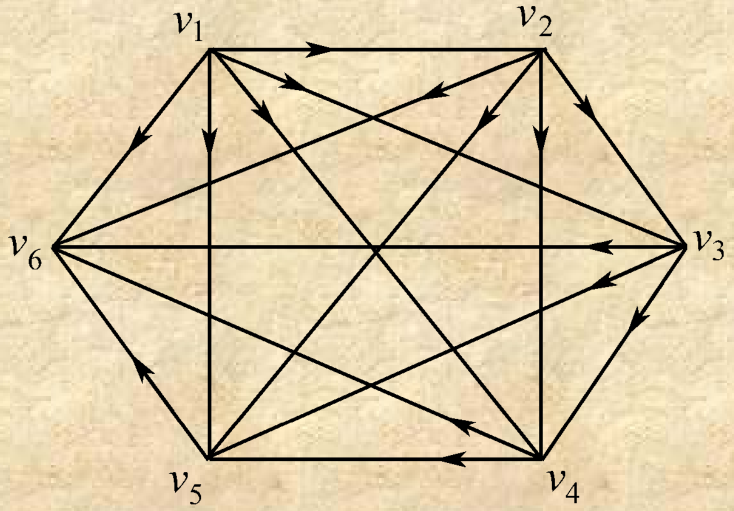

X
Y

顶点的次数

定义 （1）在无向图中，与顶点v关联的边的数目（环算两次）称为v的次数，记为$d(v)$.

（2）在有向图中，从顶点v引出的边的数目称为v的出度，记为$d^+(v)$，从顶点v引入的边的数目称为v的入度，记为$d^-(v)$，$d(v)=d^+(v)+d^-(v)$称为v的次数.





$$d(v_4) = 4$$

$$d^+(v_4) = 2$$

$$d^-(v_4) = 3$$

$$d(v_4) = 5$$

定理 1 $$\sum_{v\in V(G)} d(v) = 2\varepsilon(G)$$

推论 1 任何图中奇次顶点的总数必为偶数.

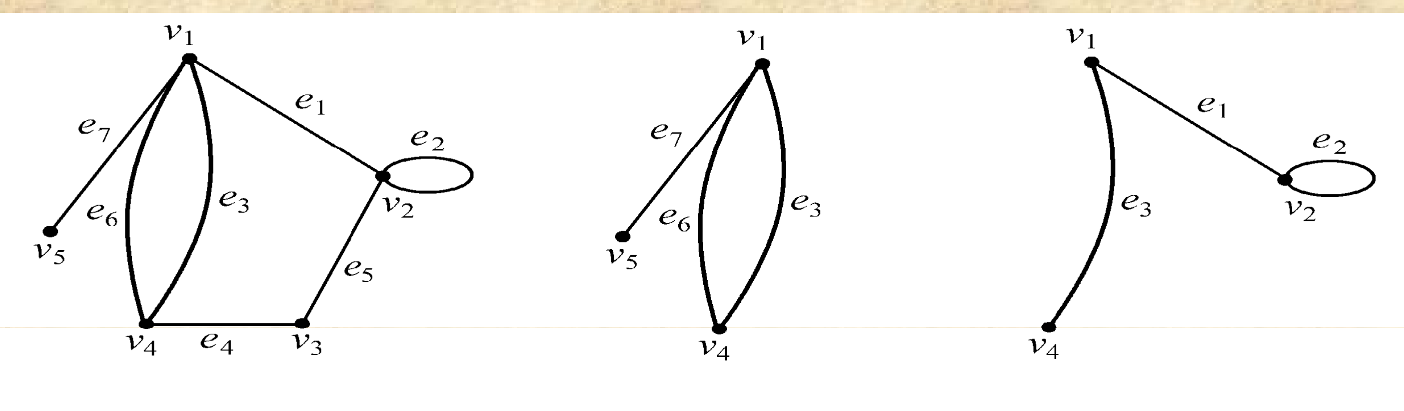
子图

定义 设图 $G=(V,E,\Psi)$,$G_1=(V_1,E_1,\Psi_1)$

(1) 若 $V_1\subseteq V$，$E_1\subseteq E$,且当 $e\in E_1$ 时，$\Psi_1(e)=\Psi(e)$,则称 G_1 是 G 的子图.特别的，若 $V_1=V$，则 G_1 称为 G 的**生成子图**.

(2) 设 $V_1\subseteq V$，且 $V_1\neq\Phi$，以 V_1 为顶点集、两个端点都在 V_1 中的图 G 的边为边集的图 G 的子图，称为 G 的由 V_1 **导出的子图**，记为 $G[V_1]$.

(3)设 $E_1\subseteq E$,且 $E_1\neq\Phi$,以 E_1 为边集,E_1 的端点集为顶点集的图 G 的子图,称为 G 的由 E_1 **导出的子图**,记为 $G[E_1]$.



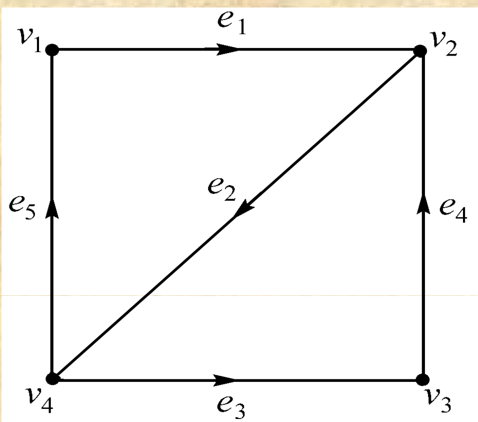
G　　$G[\{v_1,v_4,v_5\}]$　　$G[\{e_1,e_2,e_3\}]$

关联矩阵

对无向图 G，其关联矩阵 $M=(m_{ij})_{\nu\times\varepsilon}$，其中：

$$m_{ij}=\begin{cases}1 & \text{若}v_i\text{与}e_j\text{相关联}\\ 0 & \text{若}v_i\text{与}e_j\text{不关联}\end{cases}$$

注：假设图为简单图



$$M=\begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}$$

对有向图G，其关联矩阵 $M=(m_{ij})_{\nu\times\varepsilon}$，其中：

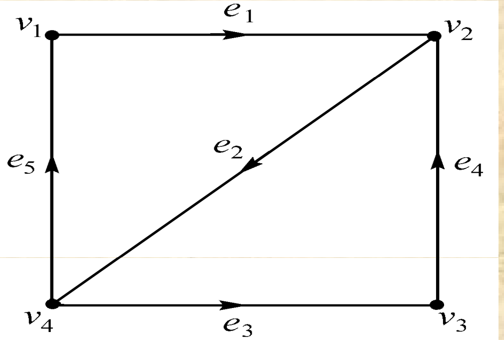
$$m_{ij}=\begin{cases}1 & \text{若}v_i\text{是}e_j\text{的起点}\\ -1 & \text{若}v_i\text{是}e_j\text{的终点}\\ 0 & \text{若}v_i\text{与}e_j\text{不关联}\end{cases}$$

邻接矩阵

对无向图 G，其邻接矩阵 $A=(a_{ij})_{\nu\times\nu}$，其中：

$$a_{ij}=\begin{cases}1 & 若v_i与v_j相邻\\ 0 & 若v_i与v_j不相邻\end{cases}$$

注：假设图为简单图



$$\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} & \\ \boldsymbol{A}=\begin{pmatrix} 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 1 & 0 & 0\\ 1 & 0 & 1 & 0 \end{pmatrix} & \begin{array}{c} v_1\\ v_2\\ v_3\\ v_4 \end{array} \end{array}$$

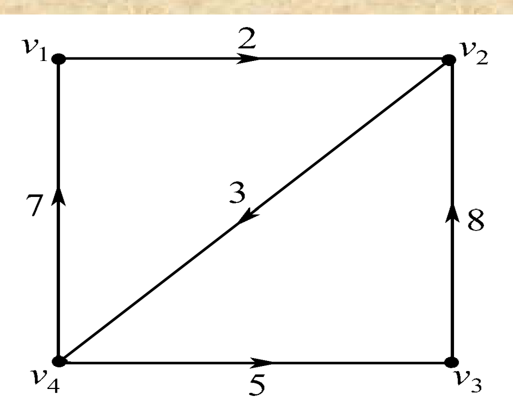
对有向图 G＝（V，E），其邻接矩阵 $A=(a_{ij})_{\nu\times\nu}$，其中：

$$a_{ij}=\begin{cases}1 & 若（v_i,\ v_j）\in E\\ 0 & 若（v_i,\ v_j）\notin E\end{cases}$$

对有向赋权图 G，其邻接矩阵 $A=(a_{ij})_{\nu\times\nu}$，其中：

$$a_{ij}=\begin{cases} w_{ij} & 若(v_i,v_j)\in E,且w_{ij}为其权 \\ 0 & 若i=j \\ \infty & 若(v_i,v_j)\notin E \end{cases}$$

无向赋权图的邻接矩阵可类似定义.



$$\begin{array}{cc} & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \mathbf{A}= & \begin{pmatrix} 0 & 2 & \infty & \infty \\ \infty & 0 & \infty & 3 \\ \infty & 8 & 0 & \infty \\ 7 & \infty & 5 & 0 \end{pmatrix} \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \end{array}$$

最短路问题及其算法

一、基本概念

二、固定起点的最短路

三、每对顶点之间的最短路

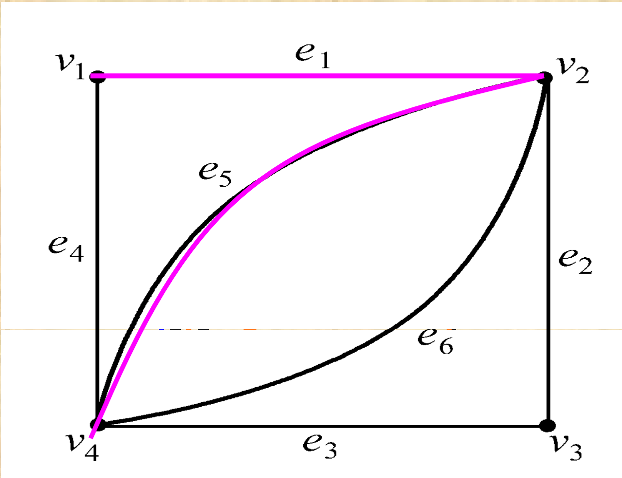
基 本 概 念

定义1　在无向图 $G=(V,E,\Psi)$中：

（1）顶点与边相互交错且 $\Psi(e_i)=v_{i-1}v_i$ $(i=1,2,\cdots,k)$的有限非空序列 $w=(v_0e_1v_1e_2\cdots v_{k-1}e_kv_k)$ 称为一条从 v_0 到 v_k 的**通路**，记为 $W_{v_0v_k}$

（2）边不重复但顶点可重复的通路称为**道路**，记为 $T_{v_0v_k}$

（3）边与顶点均不重复的通路称为**路径**，记为 $P_{v_0v_k}$



通路 $W_{v_1v_4}=v_1e_4v_4e_5v_2e_1v_1e_4v_4$

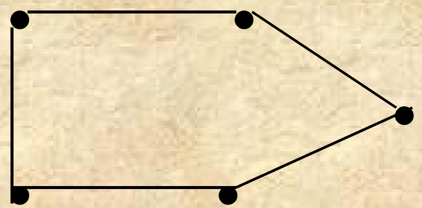
道路 $T_{v_1v_4}=v_1e_1v_2e_5v_4e_6v_2e_2v_3e_3v_4$

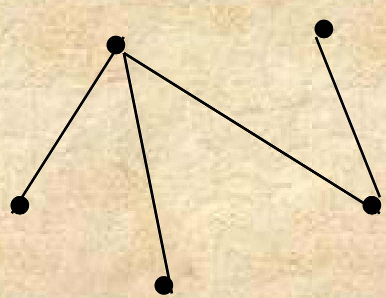
路径 $P_{v_1v_4}=v_1e_1v_2e_5v_4$

定义 2　（1）任意两点均有路径的图称为**连通图**.

（2）起点与终点重合的路径称为**圈**.

（3）连通而无圈的图称为**树**.





定义 3　（1）设 $P(u,v)$ 是赋权图 G 中从 u 到 v 的路径，

则称 $w(P)=\sum_{e\in E(P)} w(e)$ 为路径 P 的**权**.

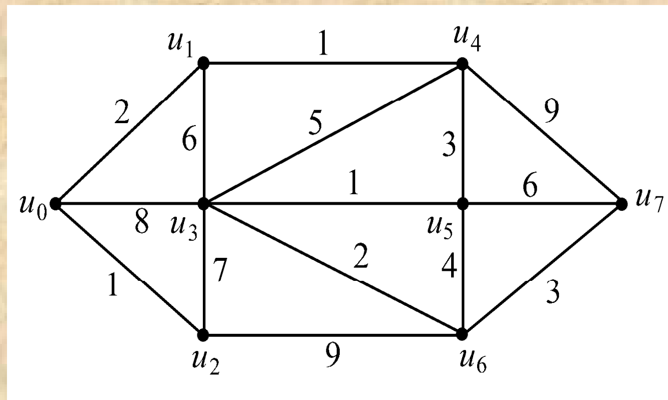
(2)　在赋权图 G 中，从顶点 u 到顶点 v 的具有最小权的路

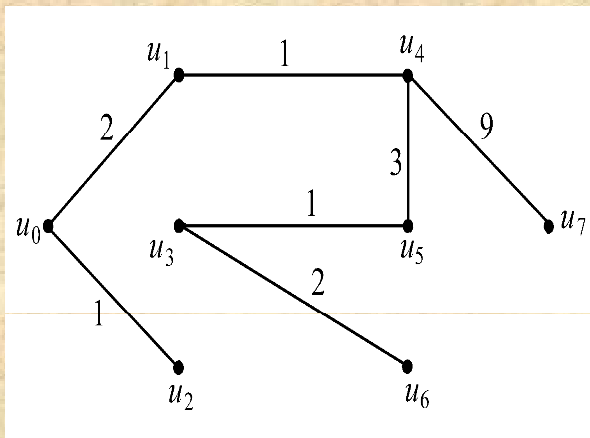
$P^*(u,v)$，称为 u 到 v 的**最短路**.

固 定 起 点 的 最 短 路

最短路是一条路径，且最短路的任一段也是最短路.

假设在u_0-v_0的最短路中只取一条，则从u_0到其余顶点的最短路将构成一棵以u_0为根的树.





因此，可采用树生长的过程来求指定顶点到其余顶点的最短路.

Dijkstra 算法：求 G 中从顶点 u_0 到其余顶点的最短路.

设 G 为赋权有向图或无向图，G 边上的权均非负.

对每个顶点，定义两个标记（$l(v)$，$z(v)$），其中:

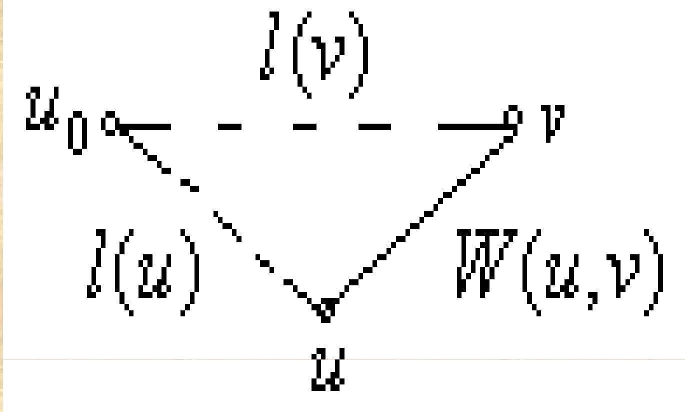
$l(v)$：表从顶点 u_0 到 v 的一条路的权.

$z(v)$：v 的父亲点，用以确定最短路的路线

算法的过程就是在每一步改进这两个标记，使最终 $l(v)$ 为从顶点 u_0 到 v 的最短路的权.

S：具有永久标号的顶点集

输入: G 的带权邻接矩阵 $w(u,v)$



算法步骤：

（1）赋初值：令 $S=\{u_0\}$，$l(u_0)=0$

$\forall v \in \overline{S} = V \setminus S$，令 $l(v) = W(u_0, v)$，$z(v) = u_0$

$u \leftarrow u_0$

（2）更新 $l(v)$、$z(v)$：　$\forall v \in \overline{S} = V \setminus S$，若 $l(v) > l(u) + W(u,v)$

则令 $l(v) = l(u) + W(u,v)$，$z(v) = u$

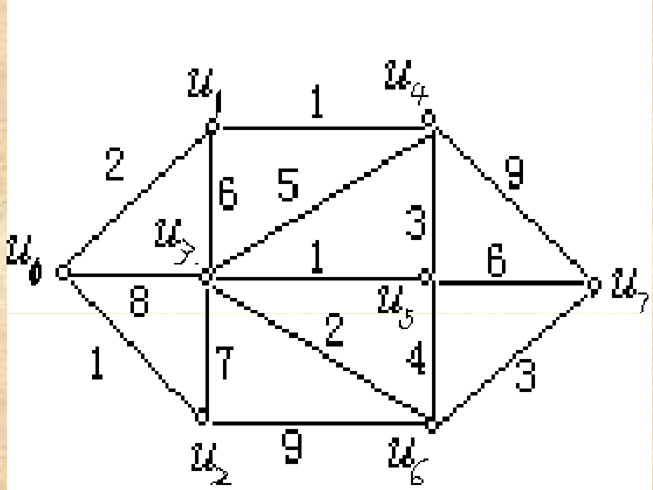
（3）　设 v^* 是使 $l(v)$ 取最小值的 $\overline{S}$ 中的顶点，则令 $S=S\cup\{v^*\}$，

$u \leftarrow v^*$

（4）　若 $\overline{S} \neq \Phi$，转 2，否则，停止.

用上述算法求出的 $l(v)$ 就是 u_0 到 v 的最短路的权，从 v 的父亲标记 $z(v)$ 追溯到 u_0，就得到 u_0 到 v 的最短路的路线.

例　求下图从顶点 u_1 到其余顶点的最短路. TO MATLAB(road1)

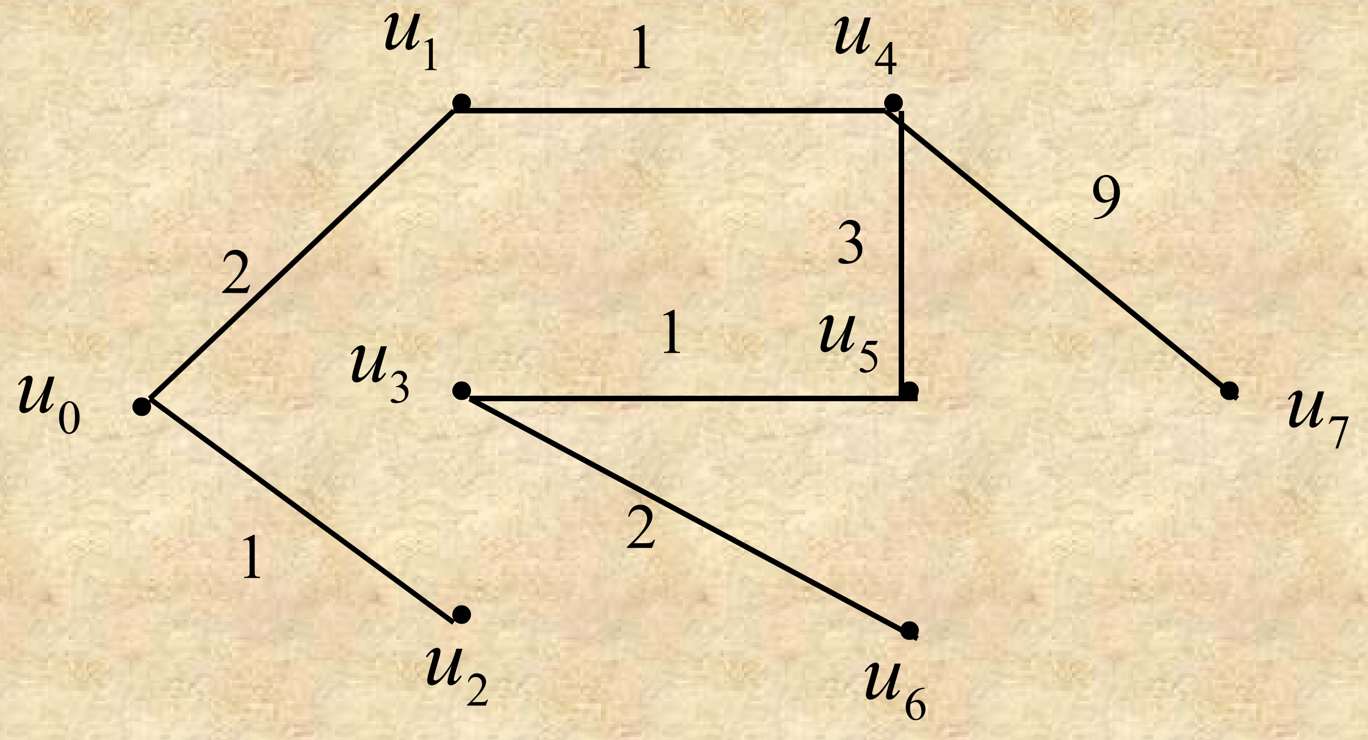


先写出带权邻接矩阵：

$$W=\begin{pmatrix} 0 & 2 & 1 & 8 & \infty & \infty & \infty & \infty \\ & 0 & \infty & 6 & 1 & \infty & \infty & \infty \\ & & 0 & 7 & \infty & \infty & 9 & \infty \\ & & & 0 & 5 & 1 & 2 & \infty \\ & & & & 0 & 3 & \infty & 9 \\ & & & & & 0 & 4 & 6 \\ & & & & & & 0 & 3 \\ & & & & & & & 0 \end{pmatrix}$$

因 G 是无向图，故 $\boldsymbol{W}$ 是对称矩阵.

迭代次数	$l(u_i)$							
	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7
1	$\boxed{0}$	∞	∞	∞	∞	∞	∞	∞
2		2	$\boxed{1}$	8	∞	∞	∞	∞
3		$\boxed{2}$		8	∞	∞	10	∞
4				8	$\boxed{3}$	∞	10	∞
5				8		$\boxed{6}$	10	12
6				$\boxed{7}$			10	12
7							$\boxed{9}$	12
8								$\boxed{12}$
最后标记:								
$l(v)$	0	2	1	7	3	6	9	12
$z(v)$	u_0	u_0	u_0	u_5	u_1	u_4	u_3	u_4



u_1
1
u_4
9
3
2
1
u_5
u_3
u_0
u_7
2
1
u_2
u_6

每 对 顶 点 之 间 的 最 短 路

（一）算法的基本思想

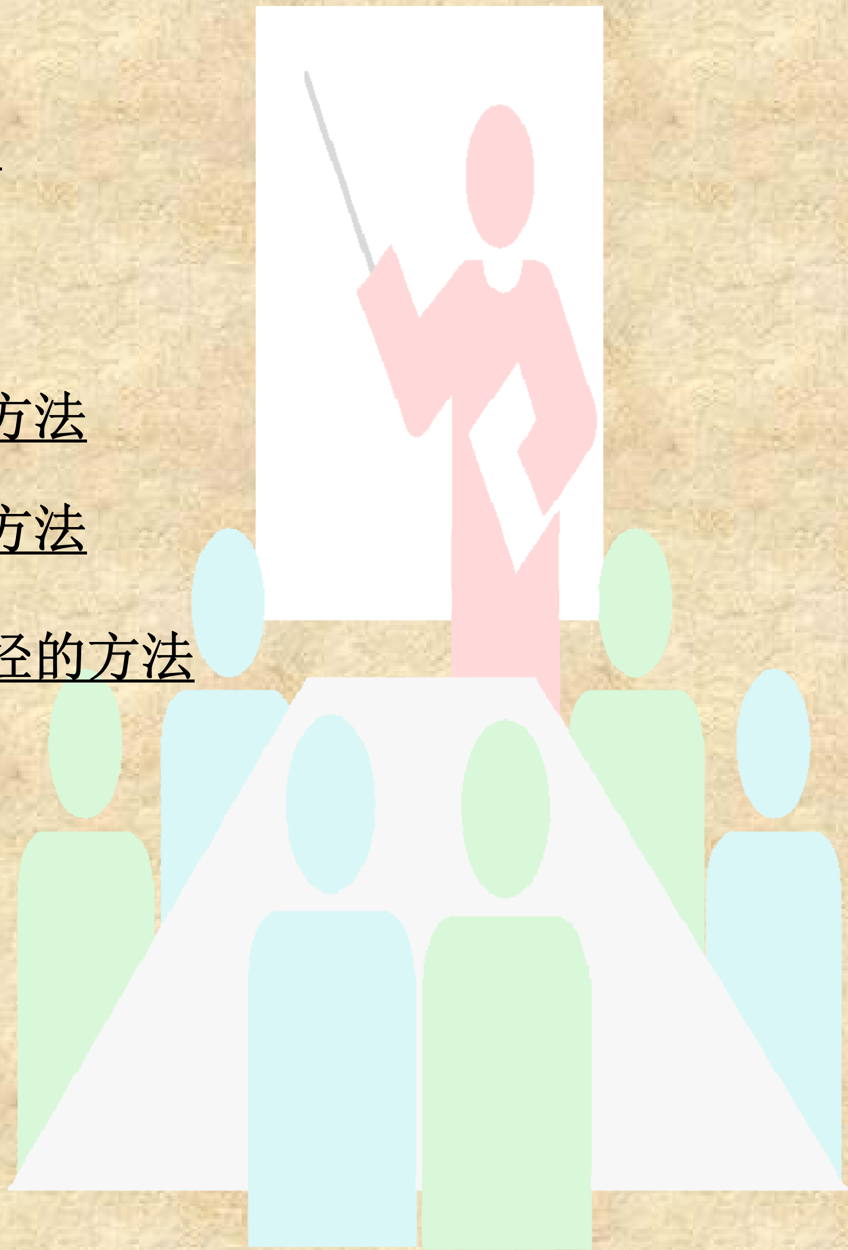
(二)算法原理

1．求距离矩阵的方法

2．求路径矩阵的方法

3．查找最短路路径的方法

（三）算法步骤



算法的基本思想

直接在图的带权邻接矩阵中用插入顶点的方法依次构造出 ν 个矩阵 $\boldsymbol{D}^{(1)}$、 $\boldsymbol{D}^{(2)}$、… 、$\boldsymbol{D}^{(\nu)}$，使最后得到的矩阵 $\boldsymbol{D}^{(\nu)}$成为图的距离矩阵，同时也求出插入点矩阵以便得到两点间的最短路径.

算法原理—— 求距离矩阵的方法

把带权邻接矩阵 $\boldsymbol{W}$ 作为距离矩阵的初值，即 $\boldsymbol{D}^{(0)}=(d_{ij}^{(0)})_{\nu\times\nu}=\boldsymbol{W}$

（1）$\boldsymbol{D}^{(1)}=(d_{ij}^{(1)})_{\nu\times\nu}$，其中 $d_{ij}^{(1)}=\min\{d_{ij}^{(0)},d_{i1}^{(0)}+d_{1j}^{(0)}\}$

$d_{ij}^{(1)}$ 是从 v_i 到 v_j 的只允许以 v_1 作为中间点的路径中最短路的长度.

（2）$\boldsymbol{D}^{(2)}=(d_{ij}^{(2)})_{\nu\times\nu}$，其中 $d_{ij}^{(2)}=\min\{d_{ij}^{(1)},d_{i2}^{(1)}+d_{2j}^{(1)}\}$

$d_{ij}^{(2)}$ 是从 v_i 到 v_j 的只允许以 v_1 、 v_2 作为中间点的路径中最短路的长度.

…

（ν）$\boldsymbol{D}^{(\nu)}=(d_{ij}^{(\nu)})_{\nu\times\nu}$，其中 $d_{ij}^{(\nu)}=\min\{d_{ij}^{(\nu-1)},d_{i\nu}^{(\nu-1)}+d_{\nu j}^{(\nu-1)}\}$

$d_{ij}^{(\nu)}$ 是从 v_i 到 v_j 的只允许以 v_1、v_2、…、v_ν 作为中间点的路径中最短路的长度. 即是从 v_i 到 v_j 中间可插入任何顶点的路径中最短路的长，因此 $\boldsymbol{D}^{(\nu)}$ 即是距离矩阵.

算法原理—— 求路径矩阵的方法

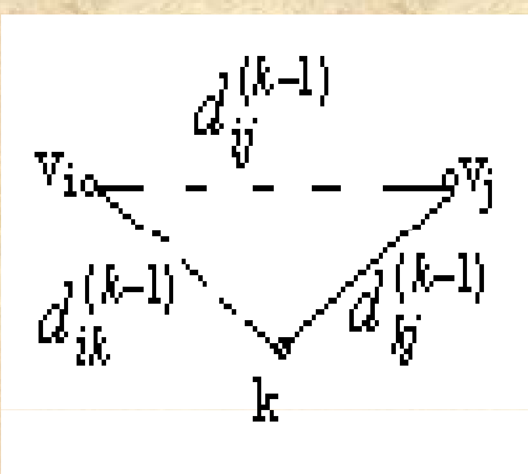
在建立距离矩阵的同时可建立路径矩阵$\boldsymbol{R}$.

$\boldsymbol{R}=(r_{ij})_{\nu\times\nu}$, r_{ij} 的含义是从 v_i 到 v_j 的最短路要经过点号为 r_{ij} 的点.

$$R^{(0)}=(r_{ij}^{(0)})_{\nu\times\nu},\quad r_{ij}^{(0)}=j$$

每求得一个 $D^{(k)}$时，按下列方式产生相应的新的 $R^{(k)}$

$$r_{ij}^{(k)}=\begin{cases} k & 若 d_{ij}^{(k-1)}>d_{ik}^{(k-1)}+d_{kj}^{(k-1)} \\ r_{ij}^{(k-1)} & 否则 \end{cases}$$



即当 v_k被插入任何两点间的最短路径时，被记录在$R^{(k)}$中，依次求$D^{(\nu)}$时求得$R^{(\nu)}$，可由$R^{(\nu)}$来查找任何点对之间最短路的路径.

算法原理—— 查找最短路路径的方法

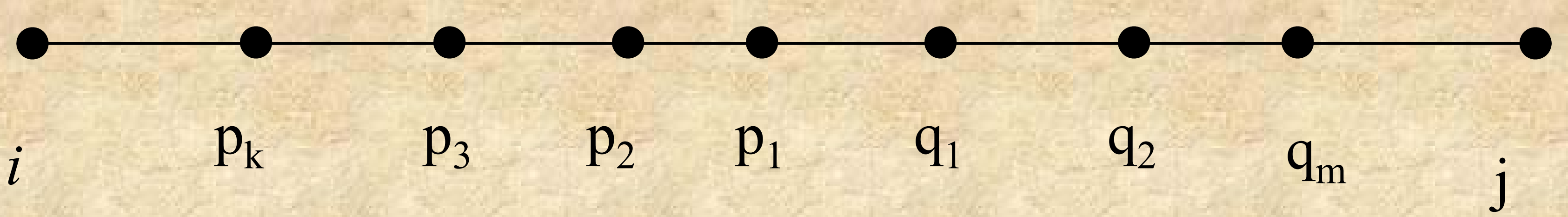
若 $r_{ij}^{(\nu)} = p_1$，则点 p_1 是点 i 到点 j 的最短路的中间点.

然后用同样的方法再分头查找．若：

（1）向点 i 追溯得：$r_{ip_1}^{(\nu)} = p_2, r_{ip_2}^{(\nu)} = p_3, \cdots, r_{ip_k}^{(\nu)} = i$

（2）向点 j 追溯得：$r_{p_1 j}^{(\nu)} = q_1, r_{q_1 j}^{(\nu)} = q_2, \cdots, r_{q_m j}^{(\nu)} = j$

则由点i到j的最短路的路径为：　$i, p_k, \cdots, p_2, p_1, q_1, q_2, \cdots, q_m, j$



算法步骤

Floyd 算法：求任意两点间的最短路.

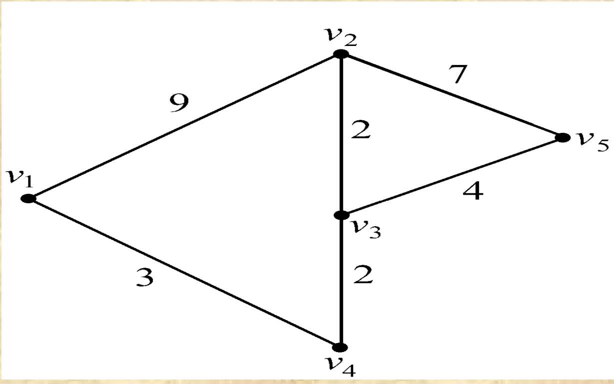
$D(i,j)$：i 到 j 的距离.
$R(i,j)$：i 到 j 之间的插入点.
输入: 带权邻接矩阵 $w(i,j)$

（1）赋初值：
对所有 i,j, $d(i,j) \leftarrow w(i,j)$, $r(i,j) \leftarrow j$, $k \leftarrow 1$

(2) 更新 $d(i,j)$, $r(i,j)$
对所有 i,j，若 $d(i,k)+d(k,j)<d(i,j)$，则 $d(i,j) \leftarrow d(i,k)+d(k,j)$, $r(i,j) \leftarrow k$

(3) 若 $k=\nu$，停止．否则 $k \leftarrow k+1$，转（2）.

例 求下图中加权图的任意两点间的距离与路径.



TO MATLAB (road2(floyd))

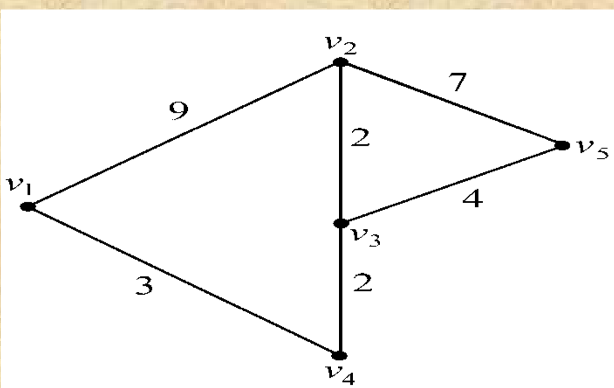
$$D=\begin{pmatrix} 0 & 7 & 5 & 3 & 9 \\ 7 & 0 & 2 & 4 & 6 \\ 5 & 2 & 0 & 2 & 4 \\ 3 & 4 & 2 & 0 & 6 \\ \boxed{9} & 6 & 4 & 6 & 0 \end{pmatrix},\ R=\begin{pmatrix} 1 & 4 & 4 & 4 & 4 \\ 4 & 2 & 3 & 3 & 3 \\ 4 & 2 & 3 & 4 & 5 \\ \boxed{1} & 3 & 3 & 4 & 3 \\ \boxed{4} & 3 & \boxed{3} & \boxed{3} & 5 \end{pmatrix}$$

$d_{51}=9$，故从 v_5 到 v_1 的最短路为 9 .

$r_{51}=4$. 由 v_4 向 v_5 追溯： $r_{54}=3, r_{53}=3$;

由 v_4 向 v_1 追溯： $r_{41}=1$

所以从 v_5 到 v_1 的最短路径为： $5\to 3\to 4\to 1$.



$$D^{(0)} = \begin{matrix} 0 & 9 & \text{Inf} & 3 & \text{Inf} \\ 9 & 0 & 2 & \text{Inf} & 7 \\ \text{Inf} & 2 & 0 & 2 & 4 \\ 3 & \text{Inf} & 2 & 0 & \text{Inf} \\ \text{Inf} & 7 & 4 & \text{Inf} & 0 \end{matrix}$$

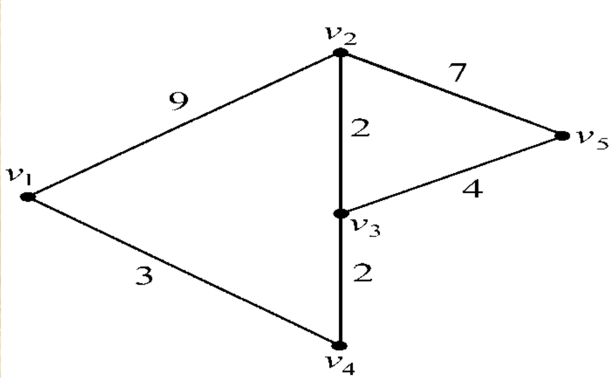
$$R^{(0)} = \begin{matrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{matrix}$$

$D^{(0)} =$

0	9	Inf	3	Inf
9	0	2	Inf	7
Inf	2	0	2	4
3	Inf	2	0	Inf
Inf	7	4	Inf	0

$R^{(0)} =$

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

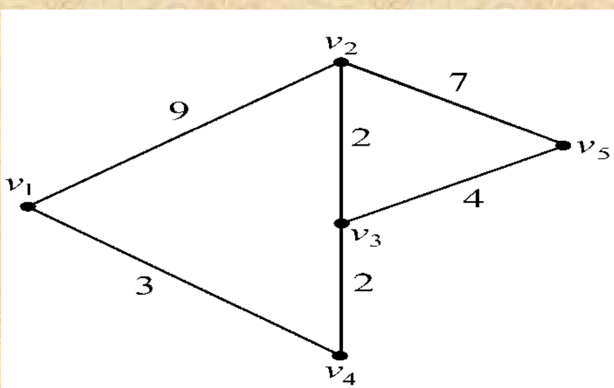


$D^{(1)} =$

0	9	Inf	3	Inf
9	0	2	12	7
Inf	2	0	2	4
3	12	2	0	Inf
Inf	7	4	Inf	0

$R^{(1)} =$

1	2	3	4	5
1	2	3	1	5
1	2	3	4	5
1	1	3	4	5
1	2	3	4	5



$D^{(1)} =$

0	9	Inf	3	Inf
9	0	2	12	7
Inf	2	0	2	4
3	12	2	0	Inf
Inf	7	4	Inf	0

$R^{(1)} =$

1	2	3	4	5
1	2	3	1	5
1	2	3	4	5
1	1	3	4	5
1	2	3	4	5

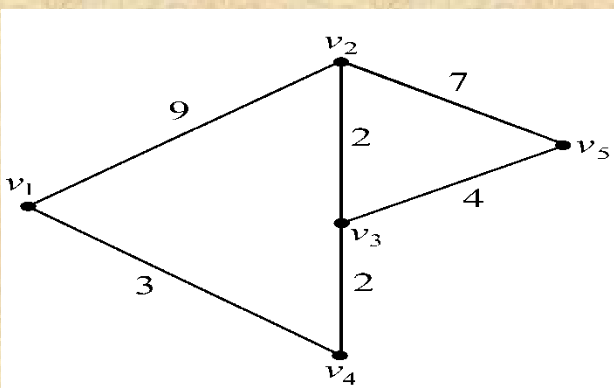
$D^{(2)} =$

0	9	11	3	16
9	0	2	12	7
11	2	0	2	4
3	12	2	0	19
16	7	4	19	0

$R^{(2)} =$

1	2	2	4	2
1	2	3	1	5
2	2	3	4	5
1	1	3	4	2
2	2	3	2	5

$$D^{(2)} = \begin{matrix} 0 & 9 & 11 & 3 & 16 \\ 9 & 0 & 2 & 12 & 7 \\ 11 & 2 & 0 & 2 & 4 \\ 3 & 12 & 2 & 0 & 19 \\ 16 & 7 & 4 & 19 & 0 \end{matrix} \qquad R^{(2)} = \begin{matrix} 1 & 2 & 2 & 4 & 2 \\ 1 & 2 & 3 & 1 & 5 \\ 2 & 2 & 3 & 4 & 5 \\ 1 & 1 & 3 & 4 & 2 \\ 2 & 2 & 3 & 2 & 5 \end{matrix}$$



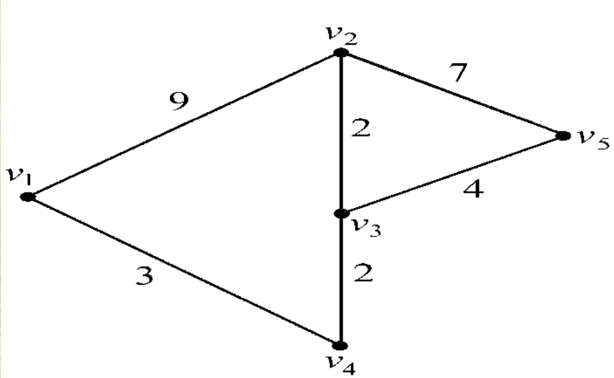
$$D^{(3)} = \begin{matrix} 0 & 9 & 11 & 3 & 15 \\ 9 & 0 & 2 & 4 & 6 \\ 11 & 2 & 0 & 2 & 4 \\ 3 & 4 & 2 & 0 & 6 \\ 15 & 6 & 4 & 6 & 0 \end{matrix} \qquad R^{(3)} = \begin{matrix} 1 & 2 & 2 & 4 & 3 \\ 1 & 2 & 3 & 3 & 3 \\ 2 & 2 & 3 & 4 & 5 \\ 1 & 3 & 3 & 4 & 3 \\ 3 & 3 & 3 & 3 & 5 \end{matrix}$$

$D^{(3)} =$

0	9	11	3	15
9	0	2	4	6
11	2	0	2	4
3	4	2	0	6
15	6	4	6	0

$R^{(3)} =$

1	2	2	4	3
1	2	3	3	3
2	2	3	4	5
1	3	3	4	3
3	3	3	3	5



$D^{(4)} =$

0	7	5	3	9
7	0	2	4	6
5	2	0	2	4
3	4	2	0	6
9	6	4	6	0

$R^{(4)} =$

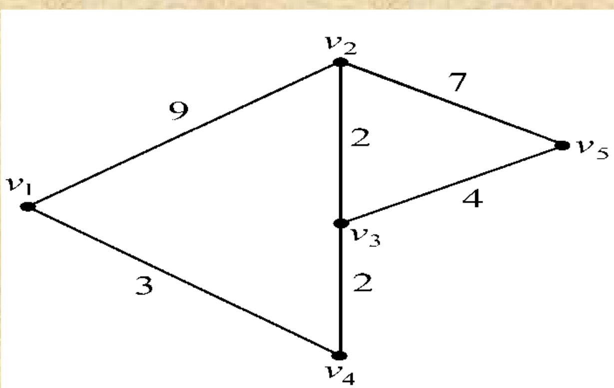
1	4	4	4	4
4	2	3	3	3
4	2	3	4	5
1	3	3	4	3
4	3	3	3	5

$D^{(4)} =$

$$\begin{matrix} 0 & \color{red}{7} & \color{red}{5} & 3 & \color{red}{9} \\ \color{red}{7} & 0 & 2 & 4 & 6 \\ \color{red}{5} & 2 & 0 & 2 & 4 \\ 3 & 4 & 2 & 0 & 6 \\ \color{red}{9} & 6 & 4 & 6 & 0 \end{matrix}$$

$R^{(4)} =$

$$\begin{matrix} 1 & \color{red}{4} & \color{red}{4} & 4 & \color{red}{4} \\ \color{red}{4} & 2 & 3 & 3 & 3 \\ \color{red}{4} & 2 & 3 & 4 & 5 \\ 1 & 3 & 3 & 4 & 3 \\ \color{red}{4} & 3 & 3 & 3 & 5 \end{matrix}$$



$D^{(5)} =$

$$\begin{matrix} 0 & 7 & 5 & 3 & 9 \\ 7 & 0 & 2 & 4 & 6 \\ 5 & 2 & 0 & 2 & 4 \\ 3 & 4 & 2 & 0 & 6 \\ 9 & 6 & 4 & 6 & 0 \end{matrix}$$

$R^{(5)} =$

$$\begin{matrix} 1 & 4 & 4 & 4 & 4 \\ 4 & 2 & 3 & 3 & 3 \\ 4 & 2 & 3 & 4 & 5 \\ 1 & 3 & 3 & 4 & 3 \\ 4 & 3 & 3 & 3 & 5 \end{matrix}$$

最 短 路 的 应 用

一、 可化为最短路问题的多阶段决策问题

二、 选 址 问 题

1． 中心问题

2． 重心问题

可化为最短路问题的多阶段决策问题

例 1 设备更新问题：企业使用一台设备，每年年初，企业领导就要确定是购置新的，还是继续使用旧的.若购置新设备，就要支付一定的购置费用；若继续使用，则需支付一定的维修费用.现要制定一个五年之内的设备更新计划，使得五年内总的支付费用最少.

已知该种设备在每年年初的价格为：

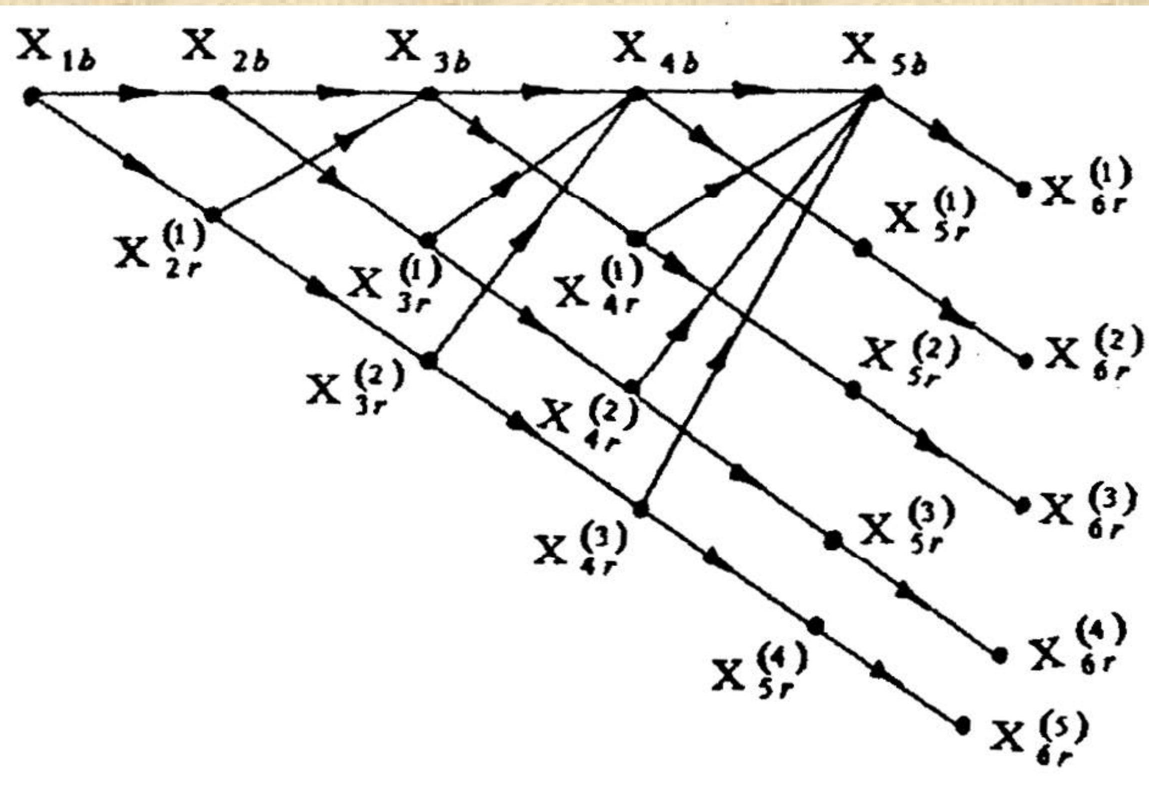
第一年	第二年	第三年	第四年	第五年
11	11	12	12	13

使用不同时间设备所需维修费为：

使用年限	0—1	1—2	2—3	3—4	4—5
维修费	5	6	8	11	18

构造加权有向图 $G1(V,\mathrm{E})$

（1）顶点集 $V=\{X_{ib},\ i=1,2,3,4,5\}\cup\{X_{ir}^{(k)},\ i=2,3,4,5,6;\ k=1,2,\cdots,i-1\}$，每个顶点代表年初的一种决策，其中顶点 X_{ib} 代表第 i 年初购置新设备的决策，顶点 $X_{ir}^{(k)}$ 代表第 i 年初修理用过 k 年的旧设备的决策



（2）弧集 E={$(X_{ib},X_{i+1,b}),(X_{ir}^{(k)},X_{i+1,b}),i$=1,2,3,4; k=1,2,⋯,i-1}

$\cup\{(X_{ib},X_{i+1,r}^{(1)}),i$=1,2,3,4,5}$\cup\{(X_{ir}^{(k)},X_{i+1,r}^{(k+1)}),i$=1,2,3,4,5;$k$=1,2,$i$-1}

若第 i 年初作了决策 X_i 后，第 i+1 年初可以作决策 X_{i+1}，则顶点 X_i 与 X_{i+1} 之间有弧(X_i,X_{i+1})，其权 W(X_i,X_{i+1})代表第 i 年初到第 i+1 年初之间的费用.例如，弧$(X_{3b},X_{4r}^{(1)})$代表第三年初买新设备，第四年初决定用第三年买的用过一年的旧设备，其权则为第三年初的购置费与第三、第四年间的维修费之和，即为 12＋5＝17.

（3）问题转化为顶点 X_{1b} 到 $X_{6r}^{(k)}$ 的最短路问题.五年的最优购置费为

$$\min_{k=1,2,3,4,5}\{d(X_{1b},X_{6r}^{(k)})\}$$

其中 $d(X_{1b},X_{6r}^{(k)})$为顶点 X_{1b} 到 $X_{6r}^{(k)}$ 的最短路的权.

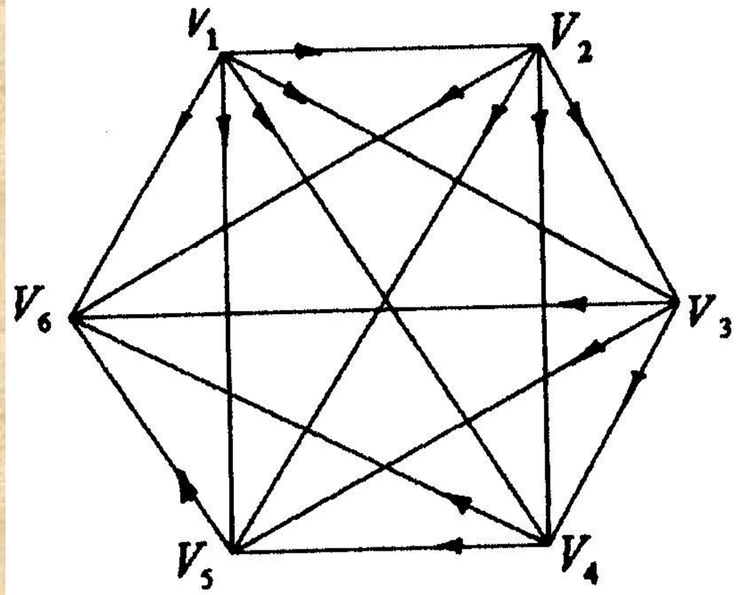
求得最短路的权为 53，而两条最短路分别为

$X_{1b}-X_{2r}^{(1)}-X_{3r}^{(2)}-X_{4b}-X_{5r}^{(1)}-X_{6r}^{(2)}$； 16+6+8+17+6=53

$X_{1b}-X_{2r}^{(1)}-X_{3b}-X_{4r}^{(1)}-X_{5r}^{(2)}-X_{6r}^{(3)}$ 16+6+17+6+8=53

因此，计划为第一、第三年初购置新设备，或第一、第四年初购置新设备，五年费用均最省，为 53.

也可构造加权有向图 $G2(V,E)$.



（1）顶点集 $V=\{V_1,V_2,V_3,V_4,V_5,V_6\}$，$V_i$ 表第 i 年初购置新设备的决策，V_6 表第五年底.

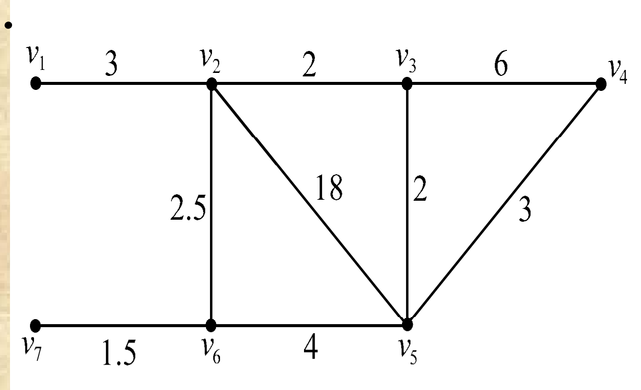
（2）弧集 $E=\{(V_i,V_j)$，$i=1,2,3,4,5;\ i<j\leqslant 6\}$，弧 (V_i,V_j) 表第 i 年初购进一台设备一直使用到第 j 年初的决策，其权 $W(V_i,V_j)$ 表由这一决策在第 i 年初到第 j 年初的总费用，如 $W(V_1,V_4)=11+5+6+8=30$.

（3）问题转化为求 V_1 到 V_6 的最短路问题，求得两条最短路为 $V_1-V_4-V_6$，$V_1-V_3-V_6$，权为 53，与图 G1(V,E)的解相同.

选址问题--中心问题

例 2　某城市要建立一个消防站，为该市所属的七个区服务，如图所示．问应设在哪个区，才能使它至最远区的路径最短．

（1）用 Floyd 算法求出距离矩阵 $\boldsymbol{D}=(d_{ij})_{\nu\times\nu}$.



（2）　计算在各点 v_i 设立服务设施的最大服务距离 $S(v_i)$.

$$S(v_i)=\max_{1\le j\le \nu}\{d_{ij}\} \qquad i=1,2,\cdots,\nu$$

（3）求出顶点 v_k，使 $S(v_k)=\min\limits_{1\le i\le \nu}\{S(v_i)\}$.

则 v_k 就是要求的建立消防站的地点．此点称为图的**中心点**．

TO　MATLAB　(road3(floyd))

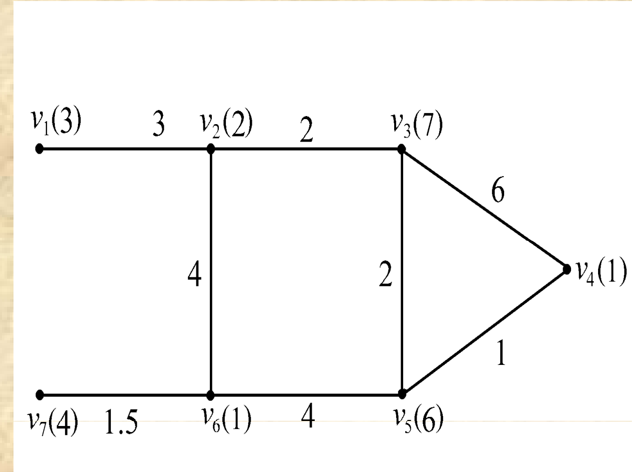
$$D=\begin{pmatrix} 0 & 3 & 5 & 10 & 7 & 5.5 & 7 \\ 3 & 0 & 2 & 7 & 4 & 2.5 & 4 \\ 5 & 2 & 0 & 5 & 2 & 4.5 & 6 \\ 10 & 7 & 5 & 0 & 3 & 7 & 8.5 \\ 7 & 4 & 2 & 3 & 0 & 4 & 5.5 \\ 5.5 & 2.5 & 4.5 & 7 & 4 & 0 & 1.5 \\ 7 & 4 & 6 & 8.5 & 5.5 & 1.5 & 0 \end{pmatrix}$$

$S(v_1)=10$, $S(v_2)=7$, $S(v_3)=6$, $S(v_4)=10$,
$S(v_5)=7$, $S(v_6)=7$, $S(v_7)=8.5$

$S(v_3)=6$,故应将消防站设在v_3处.

选址问题--重心问题

例 3 某矿区有 7 个矿点，如图所示．已知各矿点每天的产矿量为 $q(v_j)$（标在图的各顶点上）．现要从这 7 个矿点选一个来建造矿厂．问应选在哪个矿点，才能使各矿点所产的矿运到选矿厂所在地的总运力（千吨公里）最小．



（1）求距离阵 $\boldsymbol{D}=(d_{ij})_{v\times v}$.

（2） 计算各顶点作为选矿厂的总运力 $m(v_i)$

$$m(v_i)=\sum_{j=1}^{v} q(v_j)\times d_{ij} \qquad i=1,2,\cdots,v$$

（3） 求 v_k 使 $m(v_k)=\min\limits_{1\le i\le v}\{m(v_i)\}$，则 v_k 就是选矿厂应选的矿点．此点称为图 G 的**重心**或**中位点**．